

PAT-NO: JP409022380A

DOCUMENT-IDENTIFIER: JP 09022380 A

TITLE: MULTILEVEL BUS CONNECTION TYPE MULTIPROCESSOR SYSTEM

PUBN-DATE: January 21, 1997

INVENTOR-INFORMATION:

NAME
KIMURA, MASAYUKI

ASSIGNEE-INFORMATION:

NAME	COUNTRY
NEC CORP	N/A

APPL-NO: JP07191021

APPL-DATE: July 5, 1995

INT-CL (IPC): G06F012/08, G06F015/16

ABSTRACT:

PROBLEM TO BE SOLVED: To provide a multiprocessor system capable of accelerating a read request, simplifying the circuit and device constitution and improving the transmission efficiency.

SOLUTION: Each processor module 1 is provided with plural processors 3, an intra-module bus 5 for mutually connecting them, plural intrinsic caches 4, a system bus interface controller 6 for performing the arbitration control of requests on the intramodule bus 5 and interface control between the intra-module bus 5 and a system bus 2, plural copy tags 7 for system bus snooping for performing snooping to the requests on the system bus 2 as the duplicate of respective address arrays of the plural intrinsic caches 4 and the plural copy tags 8 for intra-module bus snooping for performing snooping to the requests from the plural processors 3 issued on the intra-module bus 5 as the duplicate of the respective address arrays of the plural intrinsic caches 4.

COPYRIGHT: (C)1997,JPO

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-22380

(43) 公開日 平成9年(1997)1月21日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 12/08	3 1 0	7623-5B	G 0 6 F 12/08	3 1 0 B
15/16	3 6 0		15/16	3 6 0 Z

審査請求 有 請求項の数 3 F D (全 5 頁)

(21) 出願番号 特願平7-191021

(22) 出願日 平成7年(1995)7月5日

(71) 出願人 000004237

日本電気株式会社
東京都港区芝五丁目7番1号

(72) 発明者 木村 真行

東京都港区芝五丁目7番1号 日本電気株式会社内

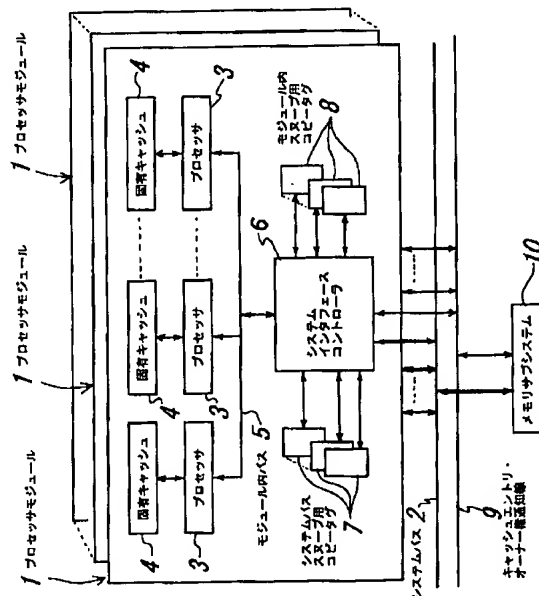
(74) 代理人 弁理士 原田 信市

(54) 【発明の名称】 マルチレベルバス結合型マルチプロセッサシステム

(57) 【要約】

【目的】 リードリクエストの高速化、回路・装置構成の簡易化、伝送効率の向上が図れるマルチプロセッサシステムを提供する。

【構成】 各プロセッサモジュール1に、複数のプロセッサ3、これらを相互に接続するモジュール内バス5、複数の固有キャッシュ4、モジュール内バス5上のリクエストの調停制御及びモジュール内バス5とシステムバス2の間のインタフェース制御を行うシステムバスインタフェースコントローラ6、複数の固有キャッシュ4のそれぞれのアドレスアレイの複製としてシステムバス2上のリクエストに対するスヌープを行うための複数のシステムバススヌープ用コピータグ7、複数の固有キャッシュ4のそれぞれアドレスアレイの複製としてモジュール内バス5上に発行される複数のプロセッサ3からのリクエストに対するスヌープを行うための複数のモジュール内バススヌープ用コピータグ8を備える。



【特許請求の範囲】

【請求項1】複数のプロセッサモジュールをシステムバスにより相互に接続したマルチプロセッサシステムにおいて、各プロセッサモジュールに、

複数のプロセッサと、

これらプロセッサを相互に接続するスプリットリード方式のモジュール内バスと、

前記複数のプロセッサにより個々に制御される複数の固有キャッシュと、

前記モジュール内バスと前記システムバスの間に実装され、前記モジュール内バス上のリクエストの調停制御及び前記モジュール内バスと前記システムバスの間のインタフェース制御を行うシステムバスインタフェースコントローラと、

このシステムバスインタフェースコントローラに接続され、前記複数の固有キャッシュのそれぞれのアドレスアレイの複製として前記システムバス上のリクエストに対するスヌープを行うために使用される複数のシステムバススヌープ用コピータグと、

前記システムバスインタフェースコントローラに接続され、前記複数の固有キャッシュのそれぞれのアドレスアレイの複製として前記モジュール内バス上に発行される前記複数のプロセッサからのリクエストに対するスヌープを行うために使用される複数のモジュール内バススヌープ用コピータグと、

を設けたことを特徴とするマルチレベルバス結合型マルチプロセッサシステム。

【請求項2】システムバスインタフェースコントローラは、複数のプロセッサのいずれか一つからリードリクエストがモジュール内バスに発行された場合、複数のモジュール内バススヌープ用コピータグを検索し、モジュール内バスに接続された他のいずれのプロセッサの固有キャッシュも該当エントリを保持していなければ、システムバスに該リードリクエストを発行することを特徴とする、請求項1記載のマルチレベルバス結合型マルチプロセッサシステム。

【請求項3】システムバスインタフェースコントローラは、複数のプロセッサのいずれか一つからリードリクエストがモジュール内バスに発行された場合、複数のモジュール内バススヌープ用コピータグを検索し、モジュール内バスに接続された他のいずれかのプロセッサの固有キャッシュが該当エントリを保持していた場合には、該エントリを保持する固有キャッシュを制御するプロセッサに対し該リクエストを発行したプロセッサに対するリプライデータの転送をモジュール内バスを介して直接行うよう指示し、複数のモジュール内バススヌープ用コピータグの内容を更新し、システムバスに対しては、データ転送により生じる複数の固有キャッシュのアドレスアレイの状態変化を複数のシステムバススヌープ用コピータグに反映するためのコピータグ更新トランザクション

を発行し、複数のシステムバススヌープ用コピータグを更新することを特徴とする、請求項1記載のマルチレベルバス結合型マルチプロセッサシステム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、マルチプロセッサシステムに関し、特にバス接続されたシンメトリック・マルチプロセッサバスのキャッシュ処理方式に係わる。

【0002】

【従来の技術】この種の技術を開示した文献として次の公報がある。

(1) 特開平5-242020号公報

(2) 特開平5-100952号公報

従来のバス結合マルチプロセッサにおいては、プロセッサとキャッシュが1対1に対応し、プロセッサは、前記公報(1)に示すごとくシステムバスと直接あるいはシステムバスインタフェースコントローラを介して接続されていた。この公報(1)のようなシステムでは、あるプロセッサがリードリクエストを発行するとそのリクエストはすべてシステムバスに発行される。また、他プロセッサが発行したリクエストは、常にシステムバススヌープ用のアドレスチェック機構(通常は固有キャッシュのアドレスアレイの複製)を用いてスヌープされ、該リクエストの対象データが自モジュール内プロセッサの配下の固有キャッシュに存在すると判断された場合には、そのプロセッサ(固有キャッシュ)に割り込み、該データをシステムバスを経由して要求元にプロセッサに転送するといった手法がとられてきた。また、前記公報

(2)に示されるごとく複数のプロセッサを同一モジュール内に実装し、これらとローカルメモリをモジュール内バスで接続するマルチプロセッサシステムも存在していた。

【0003】

【発明が解決しようとする課題】従来技術の第一の問題点はシステムバスのバスビジー率にある。その理由は、前記公報(1)のような従来のマルチプロセッサシステムにおいては、あるリクエストがリードリクエストを発行すると、そのリクエスト及びそのリードリクエストに対するデータ転送がすべてシステムバスを経由して行われるため、プロセッサ台数が増加するとシステムバスのビジー率が増加し、実効性能が大幅に低下する点にある。これに対し公報(2)のようなローカルメモリ方式も提案されており、科学技術分野のアプリケーションのように命令及びデータの局所性をソフトウェアが保証することができる場合には、リードリクエストを各プロセッサエレメント内で処理することができるためシステムバスのビジー率を低減することができ有効である。しかしながら、事務処理計算分野のアプリケーションでは、命令及びデータの局所性をソフトウェアが保証することは技術的に困難であり、システムバスのビジー率を低減

する効果は期待できない。また、ローカルメモリの容量を実アプリケーションの動作に耐え得る程の大きさに確保できない場合には、システムバスのビジー率低減効果はほとんど期待できない。

【0004】また、従来技術の第二の問題点はそのハードウェア量にある。その理由は、前記公報(1)のような従来のマルチプロセッサシステムにおいては、システム中のプロセッサ台数を増加させようとするとプロセッサエレメントの枚数を増加させるしかなく、プロセッサ台数に比例してハードウェア量が増加しかつシステムバスの物理的な長さが増加し電気特性が劣化するというこ

とである。
【0005】本発明の目的は、リードリクエストの高速化、回路・装置構成の簡易化、伝送効率の向上を図ることにある。すなわち、個々のプロセッサの負荷が上がり固有キャッシュのミス率が増加しても、システムバスのビジー率を極度に増加させることなくリードリクエストの応答を高速に行い、マルチプロセッサシステムの性能向上を図ることにある。

【0006】

【課題を解決するための手段】本発明のマルチプロセッサシステムは、複数のプロセッサモジュールをシステムバスにより相互に接続したマルチプロセッサシステムにおいて、図1に例示するように、各プロセッサモジュール1に次のような手段を備えたことを特徴とする。

- ① 複数のプロセッサ3。
- ② これらプロセッサ3を相互に接続するスプリットリード方式のモジュール内バス5。
- ③ 複数のプロセッサ3により個々に制御される複数の固有キャッシュ4。
- ④ モジュール内バス5とシステムバス2の間に実装され、モジュール内バス5上のリクエストの調停制御及びモジュール内バス5とシステムバス2の間のインタフェース制御を行うシステムバスインタフェースコントローラ6。
- ⑤ このシステムバスインタフェースコントローラ6に接続され、複数の固有キャッシュ4のそれぞれのアドレスアレイの複製としてシステムバス2上のリクエストに対するスヌープを行うために使用される複数のシステムバススヌープ用コピータグ7。
- ⑥ システムバスインタフェースコントローラ6に接続され、複数の固有キャッシュ4のそれぞれアドレスアレイの複製としてモジュール内バス5上に発行される複数のプロセッサ3からのリクエストに対するスヌープを行うために使用される複数のモジュール内バススヌープ用コピータグ8。

【0007】

【作用】本発明のマルチプロセッサにおいては、複数のプロセッサのいずれか一つからリードリクエストがモジュール内バスに発行されると、システムバスインタフェ

ースコントローラは、複数のモジュール内バススヌープ用コピータグを検索する。ここで、モジュール内バスに接続された他のいずれのプロセッサの固有キャッシュも該当エントリを保持していなかった場合には、システムバスコントローラは、システムバスに対し該リードリクエストを発行する。一方、モジュール内バスに接続された他のいずれかのプロセッサの固有キャッシュが該当エントリを保持していた場合、システムバスコントローラは、該エントリを保持する固有キャッシュを制御するプロセッサに対し該リクエストを発行したプロセッサに対するリプライデータの転送をモジュール内バスを介して直接行うよう指示し、複数のモジュール内バススヌープ用コピータグの内容を更新し、システムバスに対してはデータ転送により生じる複数の固有キャッシュのアドレスアレイの状態変化を複数のシステムバススヌープ用コピータグに反映するためのコピータグ更新トランザクションを前記システムバスに発行し、複数のシステムバススヌープ用コピータグを更新する。

【0008】次に、本発明の実施例について図面を参照して詳細に説明する。図1に本発明の一実施例を示す。

図1において、複数のプロセッサモジュール1はシステムバス2を介して相互に接続され、マルチプロセッサシステムを構成する。各プロセッサモジュール1には、複数のプロセッサ3、個々のプロセッサ3に1対1に接続される固有キャッシュ4、複数のプロセッサ3を相互に接続するスプリットリード方式のモジュール内バス5、システムバスインタフェースコントローラ6、このシステムバスインタフェースコントローラ6に接続され、個々の固有キャッシュ4のアドレスアレイの複製としてシステムバス2上のリクエストに対するスヌープを行うために使用される複数のシステムバススヌープ用コピータグ7、システムバスインタフェースコントローラ6に接続され、個々の固有キャッシュ4のアドレスアレイの複製としてモジュール内バス5上に発行されるいずれかのプロセッサ3からのリクエストに対するスヌープを行うために使用される複数のモジュール内バススヌープ用コピータグ8が搭載される。

【0009】図1のマルチプロセッサにおいては、あるプロセッサモジュール1上の複数のプロセッサ3のいずれか一つからリードリクエストがモジュール内バス5に発行されると、システムバスインタフェースコントローラ6は、自プロセッサモジュール1内の個々のプロセッサ3の固有キャッシュ4に対応した個々のモジュール内バススヌープ用コピータグ7を検索する。ここで、モジュール内バス5に接続された他のいずれのプロセッサ3の固有キャッシュ4も該当エントリを保持していなかった場合には、該リクエストを発行したプロセッサ3の固有キャッシュ4に対応するモジュール内バススヌープ用コピータグ7の内容を更新し、システムバス2に該リードリクエストを発行し、該リクエストがシステムバス2

5

に発行された時点で、該リクエストを発行したプロセッサ3の固有キャッシュ4に対応するシステムバススヌープ用コピータグ8の内容を更新する。

【0010】システムバス2上にリードリクエストが発行されると、他のプロセッサモジュール1のシステムバスインタフェースコントローラ6は、従来技術と同様にシステムバススヌープ用コピータグ7を用いてスヌープを行う。そして、該リードリクエストの対象データが、自モジュール内プロセッサ3の配下の固有キャッシュ4に存在すると判断した場合には、キャッシュエントリ・オーナー権通知線9を使用して他プロセッサモジュール1及びメモリサブシステム10にその旨を通知するとともに、そのプロセッサの固有キャッシュに対応するモジュール内バススヌープ用コピータグ7及びシステムバススヌープ用コピータグ8の内容を更新し、そのプロセッサ（固有キャッシュ）に割り込み、該リクエストに対するリプライデータを獲得し、該リプライデータをモジュール内バス5及びシステムバス2を経由して要求元プロセッサに転送する。また、キャッシュエントリ・オーナー権通知線9により、システム中のいずれのプロセッサモジュール1内の固有キャッシュ4も該当エントリを保持していないと判断される場合には、メモリサブシステム10が該当データをシステムバス2を経由して要求元プロセッサに転送する。

【0011】一方、あるプロセッサモジュール1上の複数のプロセッサ3のいずれか一つからリードリクエストがモジュール内バス2に発行され、システムバスインタフェースコントローラ6が自プロセッサモジュール1内の個々のプロセッサ3の固有キャッシュ4に対応した個々のモジュール内バススヌープ用コピータグ7を検索したときに、モジュール内バス5に接続された他のいずれかのプロセッサ3の固有キャッシュ4が該当エントリを保持していた場合には、該エントリを保持する固有キャッシュ4を制御するプロセッサ3に対し、該リクエストを発行したプロセッサ3に対するリプライデータの転送をモジュール内バス5を介して直接行うよう指示し、該データ転送に関わるプロセッサ3の固有キャッシュ4に対応するモジュール内バススヌープ用コピータグ7の内容を更新する。また、システムバス2に対しては、該データ転送により生じる複数の固有キャッシュのアドレスアレイの状態変化をシステムバススヌープ用コピータグ8に反映するためのシステムバストランザクションをシステムバス2に発行し、システムバススヌープ用コピータグ8を更新する。

【0012】モジュール内バス5は、例えば、図2に示すごとくアドレスデータ共用バス11とコマンドバス12からなり、これらのバスのアービトレーションはリクエスト線13及びリクエスト許可線14を用いてシステムバスインタフェースコントローラ6により集中制御される。また、モジュール内バス11・12上の各プロセ

6

ッサ3にはプロセッサID、モジュール内バス11・12上に発行されるリードリクエストにはリクエストIDが付加される。ここで、リードリクエストに対するリプライをシステムバスインタフェースコントローラ6もしくはプロセッサ3がアドレスデータ共用バス11に送出する際には、同時にコマンドバス12上に該リードリクエストを発行したプロセッサ3は、該リプライデータが自分に対するリプライであることを認識できる。また、システムバスインタフェースコントローラ6がプロセッサ3に対してキャッシュデータの転送を指示する際、コマンドバス12上には対象となるプロセッサ3のプロセッサID、及びもとのリードリクエストに対するリクエストIDが送出される。各プロセッサ3はモジュール内バス11・12上のリクエストの対象プロセッサIDを常時監視しており、これにより自プロセッサ3を対象とするリクエストのみを受け付け、必要なキャッシュエントリのデータを該リードリクエストのリクエストIDとともにモジュール内バス11・12に送出することができる。

【0013】

【発明の効果】本発明の第一の効果は、他のプロセッサに対する不要な割り込みを避けながらマルチプロセッサシステムにおけるシステムバスのビジー率を低減できることにある。その理由は、あるプロセッサがモジュール内バスに発行したリードリクエストが同一モジュール内バスに接続される他のプロセッサの固有キャッシュにヒットするか否かを他のプロセッサに割り込むことなく判別することができ、かつヒットした場合には該リードリクエストをシステムバスに発行する必要がなくなるからである。

【0014】本発明の第二の効果は、簡易な構成でマルチプロセッサシステムを構成できることにある。その理由は、一つのプロセッサモジュール上に複数のプロセッサを搭載できかつ、一つのシステムバスコントローラでその複数のプロセッサを制御することができること、並びにモジュール内バスにおけるキャッシュデータ転送により生じるシステムバス用コピータグの変更をシステムバス経由で行うことにより、通常のシステムバススヌープ処理とハードウェアを共用できシステムバススヌープ用コピータグの更新制御が簡素化されることにある。

【図面の簡単な説明】

【図1】本発明の一実施例を示すブロック図である。

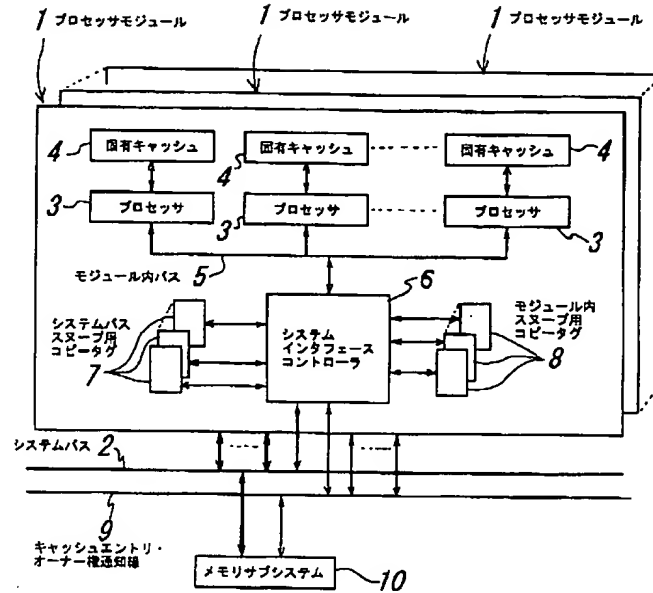
【図2】モジュール内バスの詳細を示す図である。

【符号の説明】

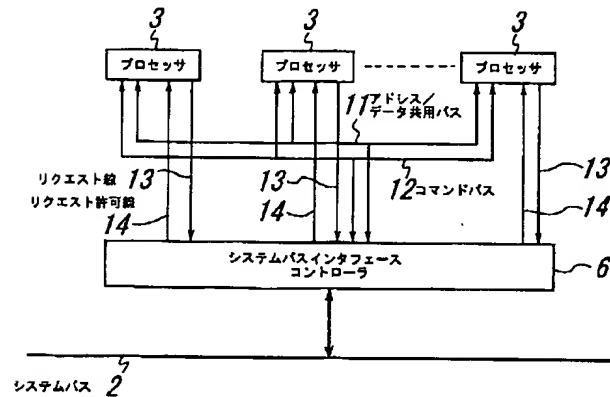
- | | |
|---|---------------------|
| 1 | プロセッサモジュール |
| 2 | システムバス |
| 3 | プロセッサ |
| 4 | 固有キャッシュ |
| 5 | モジュール内バス |
| 6 | システムバスインタフェースコントローラ |

- | | | | |
|----|--------------------|----|--------------|
| 7 | システムバススヌープ用コピータグ | 11 | アドレス/データ共用バス |
| 8 | モジュール内バススヌープ用コピータグ | 12 | コマンドバス |
| 9 | キャッシュエントリ・オーナー権通知線 | 13 | リクエスト線 |
| 10 | メモリサブシステム | 14 | リクエスト許可線 |

【図1】



【図2】



* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] A multiprocessor system which is characterized by providing the following and which connected two or more processor modules mutually by system bus To each processor module, they are two or more processors. A bus in a module of a split lead method which connects these processors mutually Two or more proper caches separately controlled by said two or more processors A system bus interface controller which is mounted between a bus in said module, and said system bus, and performs mediation control of a request on said bus in a module, and interface control between a bus in said module, and said system bus, Two or more copy tags for system bus snoops used in order to connect with this system bus interface controller and to perform a snoop to a request on said system bus as a duplicate of each address array of two or more of said proper caches, It connects with said system bus interface controller. Two or more copy tags for the bus snoops in a module used in order to perform a snoop to a request from said two or more processors published on said bus in a module as a duplicate of each address array of two or more of said proper caches

[Claim 2] A system bus interface controller is a multi-level bus joint mold multiprocessor system according to claim 1 which searches two or more copy tags for the bus snoops in a module, and is characterized by publishing this lead request to a system bus if neither of the proper caches of the processors of the others connected to a bus in a module holds an applicable entry when a lead request is published into a bus in a module from any one of two or more of the processors.

[Claim 3] A system bus interface controller When a lead request is published into a bus in a module from any one of two or more of the processors, When a proper cache of one which searched two or more copy tags for the bus snoops in a module, and was connected to a bus in a module of other processors holds an applicable entry It points so that a reply data transfer to a processor which published this request to a processor which controls a proper cache holding this entry may be directly performed through a bus in a module. Update the content of two or more copy tags for the bus snoops in a module, and a system bus is received. A renewal transaction of a copy tag for reflecting in two or more copy tags for system bus snoops a change of state of an address array of two or more proper caches produced by data transfer is published. A multi-level bus joint mold multiprocessor system according to claim 1 characterized by updating two or more copy tags for system bus snoops.

[Translation done.]

* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Industrial Application] Especially this invention relates to cache coincidence mode of processing of the symmetric multiprocessor bus by which the bus connection was carried out about a multiprocessor system.

[0002]

[Description of the Prior Art] There is the following official report as reference which indicated this kind of technology.

(1) In the bus joint multiprocessor of the JP,5-242020,A (2) JP,5-100952,A former, the processor and the cache corresponded to 1 to 1, and the processor was connected through the system bus, direct, or system bus interface controller, as shown in said official report (1). In a system like this official report (1), if a certain processor publishes a lead request, the whole of that request will be published by the system bus. Moreover, the request which other processors published A snoup is always carried out using the address check device for system bus snoups (usually duplicate of the address array of a proper cache). When the object data of this request is judged to exist in the proper cache of the subordinate of the processor in a self-module, the processor (proper cache) was interrupted and the technique of transmitting this data to a requiring agency via a system bus at a processor has been taken. Moreover, as shown in said official report (2), two or more processors were mounted in the same module, and the multiprocessor system which connects a local memory with these by bus in a module also existed.

[0003]

[Problem(s) to be Solved by the Invention] The first trouble of the conventional technology is in the rate of bus busy of a system bus. In a conventional multiprocessor system like said official report (1), if a certain request publishes a lead request, since the request and data transfer to the lead request will be altogether performed via a system bus, if the number of a processor increases, the rate of a busy of a system bus will increase, and the reason is in the point that effectiveness ability falls substantially. On the other hand, when a local memory method like an official report (2) is also proposed and software can guarantee the locality of an instruction and data like the application of a scientific and research discipline, since a lead request can be processed within each processor element, the rate of a busy of a system bus can be reduced and it is effective. However, in the application of a paperwork computational application, it is technically difficult for software to guarantee the locality of an instruction and data, and the effect of reducing the rate of a busy of a system bus cannot be expected. Moreover, when capacity of a local memory cannot be secured to magnitude to the extent that it can be equal to actuation of real application, the rate reduction effect of a busy of a system bus can hardly be expected.

[0004] Moreover, the second trouble of the conventional technology is in the amount of hardware. In a conventional multiprocessor system like said official report (1), I hear that cannot but make the number of sheets of a processor element increase when you are going to make it increase the number of a processor in a system, and the amount of hardware increases, and the physical length of a system bus increases in proportion to the number of a processor, and an electrical property deteriorates, and it is

[the] reasonable.

[0005] The object of this invention is shown in aiming at improvement in the speed of a lead request, simplification of a circuit and an equipment configuration, and improvement in transmission efficiency. That is, even if the load of each processor goes up and the rate of a mistake of a proper cache increases, a high speed is answered in a lead request, without making the rate of a busy of a system bus increase to the degree of pole, and it is shown in aiming at improvement in the engine performance of a multiprocessor system.

[0006]

[Means for Solving the Problem] A multiprocessor system of this invention is characterized by equipping each processor module 1 with the following means so that two or more processor modules may be illustrated to drawing 1 in a multiprocessor system which connected mutually by system bus.

** Two or more processors 3.

** The bus 5 in a module of a split lead method which connects these processors 3 mutually.

** Two or more proper caches 4 separately controlled by two or more processors 3.

** The system bus interface controller 6 which is mounted between the bus 5 in a module, and a system bus 2, and performs mediation control of a request on the bus 5 in a module, and interface control between the bus 5 in a module, and a system bus 2.

** Two or more copy tags 7 for system bus snoops used in order to connect with this system bus interface controller 6 and to perform a snoup to a request on a system bus 2 as a duplicate of each address array of two or more proper caches 4.

** Two or more copy tags 8 for the bus snoops in a module used in order to connect with the system bus interface controller 6 and to perform a snoup to a request from two or more processors 3 of two or more proper caches 4 published on the bus 5 in a module as a duplicate of an address array, respectively.

[0007]

[Function] In the multiprocessor of this invention, if a lead request is published into the bus in a module from any one of two or more of the processors, a system bus interface controller will search two or more copy tags for the bus snoops in a module. Here, when neither of the proper caches of the processors of the others connected to the bus in a module holds the applicable entry, a system bus controller publishes this lead request to a system bus. When the proper cache of one which was connected to the bus in a module of other processors holds the applicable entry, on the other hand, a system bus controller It points so that the reply data transfer to the processor which published this request to the processor which controls the proper cache holding this entry may be directly performed through the bus in a module. The content of two or more copy tags for the bus snoops in a module is updated. The renewal transaction of a copy tag for reflecting in two or more copy tags for system bus snoops the change of state of the address array of two or more proper caches produced by data transfer to a system bus is published to said system bus. Two or more copy tags for system bus snoops are updated.

[0008] Next, the example of this invention is explained to details with reference to a drawing. One example of this invention is shown in drawing 1 . In drawing 1 , it connects mutually through a system bus 2, and two or more processor modules 1 constitute a multiprocessor system. In each processor module 1 Two or more processors 3, It connects with the proper cache 4 connected to each processor 3 1 to 1, the bus 5 in a module of the split lead method which connects two or more processors 3 mutually, the system bus interface controller 6, and this system bus interface controller 6. It connects with two or more copy tags 7 for system bus snoops and the system bus interface controller 6 which are used in order to perform the snoup to the request on a system bus 2 as a duplicate of the address array of each proper cache 4. Two or more copy tags 8 for the bus snoops in a module used in order to perform the snoup to the request from one which is published on the bus 5 in a module as a duplicate of the address array of each proper cache 4 of the processors 3 are carried.

[0009] In the multiprocessor of drawing 1 , if a lead request is published into the bus 5 in a module from any one of two or more of the processors 3 on a certain processor module 1, the system bus interface controller 6 will search each copy tag 7 for the bus snoops in a module corresponding to the proper cache 4 of each processor 3 in the self-processor module 1. When neither of the proper caches 4 of the

processors 3 of the others connected to the bus 5 in a module holds the applicable entry here The content of the copy tag 7 for the bus snoops in a module corresponding to the proper cache 4 of the processor 3 which published this request is updated. When this lead request is published to a system bus 2 and this request is published by the system bus 2, the content of the copy tag 8 for system bus snoops corresponding to the proper cache 4 of the processor 3 which published this request is updated.

[0010] If a lead request is published on a system bus 2, the system bus interface controller 6 of other processor modules 1 will perform a snoup using the copy tag 7 for system bus snoops like the conventional technology. and when the object data of this lead request judges that it exists in the proper cache 4 of the subordinate of the processor 3 in a self-module While notifying that to the other processor modules 1 and the memory subsystem 10 using the advice line 9 of cache entry ownership The content of the copy tag 7 for the bus snoops in a module corresponding to the proper cache of the processor and the copy tag 8 for system bus snoops is updated. The processor (proper cache) is interrupted, the reply data to this request is gained, and this reply data is transmitted to a requiring agency processor via the bus 5 in a module, and a system bus 2. Moreover, by the advice line 9 of cache entry ownership, when the proper cache 4 in which [in a system] processor module 1 is also judged not to hold the applicable entry, the memory subsystem 10 transmits applicable data to a requiring agency processor via a system bus 2.

[0011] On the other hand, a lead request is published into the bus 2 in a module from any one of two or more of the processors 3 on a certain processor module 1. When the system bus interface controller 6 searches each copy tag 7 for the bus snoops in a module corresponding to the proper cache 4 of each processor 3 in the self-processor module 1 When the proper cache 4 of one which was connected to the bus 5 in a module of other processors 3 holds the applicable entry As opposed to the processor 3 which controls the proper cache 4 holding this entry It points so that the reply data transfer to the processor 3 which published this request may be directly performed through the bus 5 in a module, and the content of the copy tag 7 for the bus snoops in a module corresponding to the proper cache 4 of the processor 3 in connection with this data transfer is updated. Moreover, the system bus transaction for reflecting in the copy tag 8 for system bus snoops the change of state of the address array of two or more proper caches produced by this data transfer to a system bus 2 is published to a system bus 2, and the copy tag 8 for system bus snoops is updated.

[0012] The bus 5 in a module consists of an address-data shared bus 11 and a command bus 12, as shown in drawing 2, and centralized control of the Arbitration of these buses is carried out by the system bus interface controller 6 using the request line 13 and the request authorization line 14. Moreover, Request ID is added to the lead request published on Processor ID and the bus 11-12 in a module by each processor 3 on the bus 11-12 in a module. In case the system bus interface controller 6 or a processor 3 sends out the reply to a lead request to the address-data shared bus 11 here, the processor 3 which published this lead request on the command bus 12 simultaneously can recognize that this reply data is the reply to itself. Moreover, in case the system bus interface controller 6 directs a cache data transfer to a processor 3, on a command bus 12, the request ID to Processor ID and the lead request from the first of the target processor 3 is sent out. Each processor 3 is monitoring the object processor ID of the request on the bus 11-12 in a module continuously, can receive only the request for intraprocessor 3 by this, and can send out the data of a required cache entry to the bus 11-12 in a module with the request ID of this lead request.

[0013]

[Effect of the Invention] The first effect of this invention is to be able to reduce the rate of a busy of the system bus in a multiprocessor system, avoiding unnecessary interruption to other processors. The reason is that it becomes unnecessary that it publishes this lead request to a system bus when it can distinguish, without whether the lead request which a certain processor published into the bus in a module hits into the proper cache of other processors connected to the bus in the same module, and interrupting other processors and hits.

[0014] The second effect of this invention is to be able to constitute a multiprocessor system from a simple configuration. The reason is in being able to share the usual system bus snoup processing and

hardware, and updating control of the copy tag for system bus snoups being simplified that two or more processors can be carried on one processor module, and two or more of the processors can be controlled by one system bus controller, and by changing into a list the copy tag for system buses produced by the cache data transfer in the bus in a module via a system bus.

[Translation done.]

* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is the block diagram showing one example of this invention.

[Drawing 2] It is drawing showing the details of the bus in a module.

[Description of Notations]

- 1 Processor Module
- 2 System Bus
- 3 Processor
- 4 Proper Cache
- 5 Bus in Module
- 6 System Bus Interface Controller
- 7 Copy Tag for System Bus Snoops
- 8 Copy Tag for Bus Snoops in Module
- 9 Advice Line of Cache Entry Ownership
- 10 Memory Subsystem
- 11 Address / Data Shared Bus
- 12 Command Bus
- 13 Request Line
- 14 Request Authorization Line

[Translation done.]